Java programming 2_2

1. Update the JavaBank.java application to implement a new light blue company color that is to be used across all of the graphical user interfaces in the Java application. The values to be used are Red: 173, Green 216 and Blue 230.

```
private final int R = 173; private final int G = 216; private final int B = 230;
```

This exercise uses the bike project from Java Programming 2-1: Working with Pre-Written Code Practice Activity. If you have not completed that section, please go and do so before continuing with this exercise.

2. Create the following interface in the bike project that sets the name of the bike company as an unchangeable value. It also defines the methods that must be implemented by any class that uses the interface.

```
package bikeproject;

public interface BikeParts {

//constant declaration public final String MAKE = "Oracle Bikes";

//required methods after implementation public String getHandleBars();

public void setHandleBars(String newValue);

public String getTyres();

public void setTyres(String newValue);

public String getSeatType();

public void setSeatType(String newValue);

}//end interface BikeParts
```

3. Create an interface named MountainParts that has a constant named TERRAIN that will store the String value “off_road”. The interface will define two methods that accept a String argument name newValue and two that will return the current value of an instance field. The methods are to be named: getSuspension, setSuspension, getType , setType.

```
public interface MountainParts {

public final String TERRAIN = "off_road";

public String getSuspension();

public void setSuspension(String newValue);

public String getType();

public void setType(String newValue);

}
```

4. Create a RoadParts interface that has a constant named terrain that will store the String value “track_racing”. The interface will define two methods that accept a String argument name newValue and two that will return the current value of an instance field. The methods are to be named: getTyreWidth, setTyreWidth, getPostHeight, setPostHeight.

```
package bikeproject;
public interface RoadParts {
public final String TERRAIN = "track_racing";
public int getTyreWidth();
public void setTyreWidth(int newValue);
public int getPostHeigh();
public void setPostHeigh(int newValue);
}
```

Use the BikeParts interface with the Bike class adding any unimplemented methods required. Add the required internal code for each of the added methods.

```
@Override
public String getHandleBars() {
return handleBars;
}
@Override
public void setHandleBars(String newValue) {
this.handleBars = newValue;
}
@Override
public String getTyres() {
return tyres;
}
@Override
public void setTyres(String newValue) {
this.tyres = newValue;
}
```

```
@Override
public String getSeatType() {
return seatType;
}
@Override
public void setSeatType(String newValue) {
this.seatType = newValue;
}
```

5. Use the MountainParts interface with the MountainBike class adding any unimplemented methods required. Add the required internal code for each of the added methods.

```
@Override
public String getSuspension() {
return suspension;
}
@Override
public void setSuspension(String newValue) {
this.suspension = newValue;
}
@Override
public String getType() {
return type;
}
@Override
public void setType(String newValue) {
this.type = newValue;
}
```

6. Use the RoadParts interface with the RoadBike class adding any unimplemented methods required. Add the required internal code for each of the added methods.

```
@Override
```

```
public int getTyreWidth() {
return tyreWidth;
}
@Override
public void setTyreWidth(int newValue) {
this.tyreWidth = newValue;
}
@Override
public int getPostHeigh() {
return postHeight;
}
@Override
public void setPostHeigh(int newValue) {
this.postHeight = newValue;
}
```

7. Run and test your program, it should do exactly as it did before.

8. At the bottom of the driver class update the height of the post for bike1 to 20 instead of 22.

```
public void RoadBike() {
this("drop", "racing", "tread less", "razor", 19, 20, 20);
}
```

9. Display the values of bike1 to screen to confirm the change.

10. Run and test your program.

Final program:

JavaBank.java (with light blue company color)

```
import java.awt.Color;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
class JavaBank
```

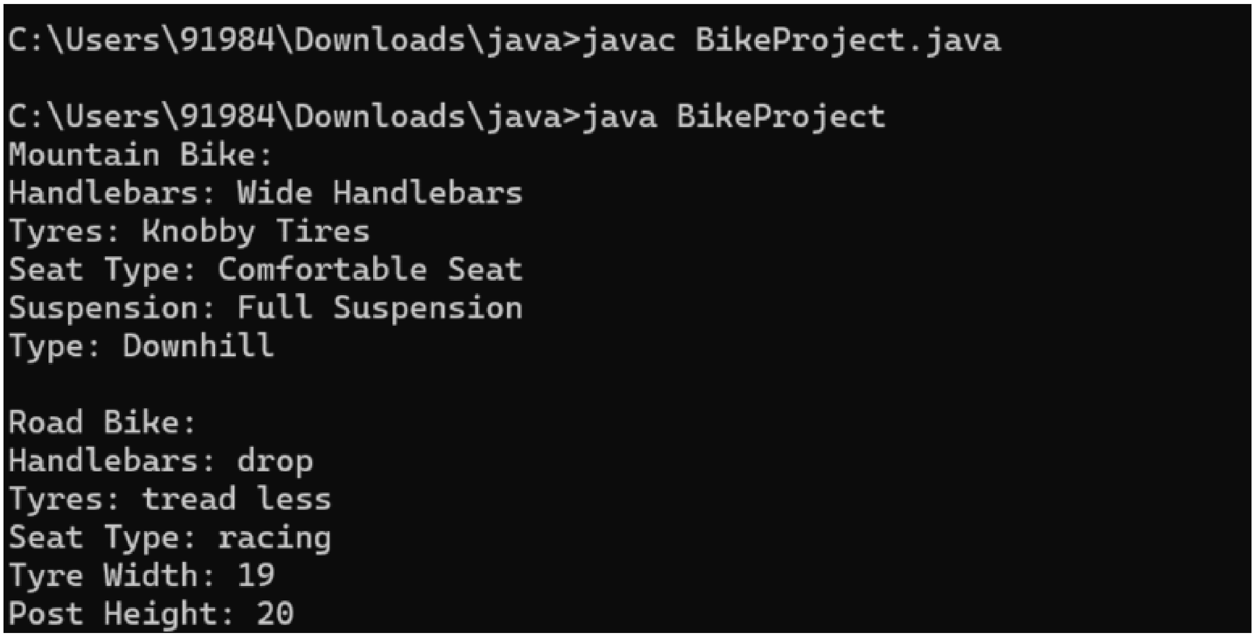
```
{
private final int R = 173;
private final int G = 216;
private final int B = 230;
public static final Color COMPANY_COLOR = new Color(173, 216, 230);
public void createGUI() {
JFrame frame = new JFrame("JavaBank");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JPanel panel = new JPanel();
panel.setBackground(COMPANY_COLOR);
JButton button = new JButton("Submit");
button.setBackground(COMPANY_COLOR);
panel.add(button);
frame.add(panel);
frame.setSize(300, 200);
frame.setVisible(true);
}
public static void main(String[] args) {
new JavaBank().createGUI();
}
}
interface BikeParts {
// Constant declaration
public final String MAKE = "Oracle Bikes";
// Required methods after implementation
public String getHandleBars();
public void setHandleBars(String newValue);
public String getTyres();
public void setTyres(String newValue);
public String getSeatType();
public void setSeatType(String newValue);
}
interface MountainParts {
// Constant declaration
public final String TERRAIN = "off_road";
// Required methods
public String getSuspension();
public void setSuspension(String newValue);
public String getType();
public void setType(String newValue);
}
interface RoadParts {
// Constant declaration
public final String TERRAIN = "track_racing";
// Required methods
public int getTyreWidth();
public void setTyreWidth(int newValue);
public int getPostHeigh();
public void setPostHeigh(int newValue);
}
class Bike implements BikeParts {
private String handleBars;
private String tyres;
private String seatType;
@Override
public String getHandleBars() {
return handleBars;
}
```

```
@Override

public void setHandleBars(String newValue) {

this.handleBars = newValue;

}

@Override

public String getTyres() {

return tyres;

}

@Override

public void setTyres(String newValue) {

this.tyres = newValue;

}

@Override

public String getSeatType() {

return seatType;

}

@Override

public void setSeatType(String newValue) {

this.seatType = newValue;

}

}

class MountainBike extends Bike implements MountainParts {

private String suspension;

private String type;

@Override

public String getSuspension() {

return suspension;

}

@Override
```

```
public void setSuspension(String newValue) {
this.suspension = newValue;
}
@Override
public String getType() {
return type;
}
@Override
public void setType(String newValue) {
this.type = newValue;
}
}
class RoadBike extends Bike implements RoadParts {
private int tyreWidth;
private int postHeight;
@Override
public int getTyreWidth() {
return tyreWidth;
}
@Override
public void setTyreWidth(int newValue) {
this.tyreWidth = newValue;
}
@Override
public int getPostHeigh() {
return postHeight;
}
@Override
public void setPostHeigh(int newValue) {
```

```
this.postHeight = newValue;
}
// Constructor to set default values
public RoadBike() {
this.setHandleBars("drop");
this.setSeatType("racing");
this.setTyres("tread less");
this.setTyreWidth(19);
this.setPostHeigh(20);
}
}
public class BikeProject {
public static void main(String[] args) {
// Creating an instance of MountainBike
MountainBike mountainBike = new MountainBike();
mountainBike.setHandleBars("Wide Handlebars");
mountainBike.setTyres("Knobby Tires");
mountainBike.setSeatType("Comfortable Seat");
mountainBike.setSuspension("Full Suspension");
mountainBike.setType("Downhill");
// Displaying MountainBike details
System.out.println("Mountain Bike:");
System.out.println("Handlebars: " + mountainBike.getHandleBars());
System.out.println("Tyres: " + mountainBike.getTyres());
System.out.println("Seat Type: " + mountainBike.getSeatType());
System.out.println("Suspension: " + mountainBike.getSuspension());
System.out.println("Type: " + mountainBike.getType());
// Creating an instance of RoadBike
RoadBike roadBike = new RoadBike();
```

```
// Displaying RoadBike details

System.out.println("\nRoad Bike:");

System.out.println("Handlebars: " + roadBike.getHandleBars());

System.out.println("Tyres: " + roadBike.getTyres());

System.out.println("Seat Type: " + roadBike.getSeatType());

System.out.println("Tyre Width: " + roadBike.getTyreWidth());

System.out.println("Post Height: " + roadBike.getPostHeigh());

}

}
```



```
C:\Users\91984\Downloads\java>javac BikeProject.java

C:\Users\91984\Downloads\java>java BikeProject
Mountain Bike:
Handlebars: Wide Handlebars
Tyres: Knobby Tires
Seat Type: Comfortable Seat
Suspension: Full Suspension
Type: Downhill

Road Bike:
Handlebars: drop
Tyres: tread less
Seat Type: racing
Tyre Width: 19
Post Height: 20
```